

CSC 341 - Fall 2022

Problem Set 5 Solutions

Problem 1. Give a context-free grammar that produces the language $\{\omega \in \mathbb{Z}_2^* \mid |\omega| \text{ is even}\}$.

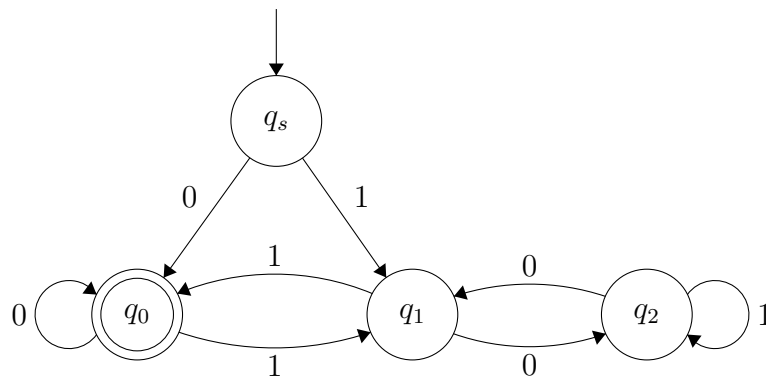
Solution 1. We define the context-free grammar with the following production rules.

- $S \rightarrow \epsilon$
- $S \rightarrow 00S$
- $S \rightarrow 01S$
- $S \rightarrow 10S$
- $S \rightarrow 11S$

Here S is the initial variable, and $0, 1$ are terminals.

Problem 2. Give a context-free grammar that produces the language $\{\omega \in \mathbb{Z}_2^* \mid 3 \text{ divides } \omega\}$, where ω is interpreted as a base 2 number.

Solution 2. A good way to come up with a context-free grammar for a regular language (which this one is) is to write down a DFA for it and then convert it into the grammar. The DFA below accepts the language by tracking the residue of our input modulo 3 as it processes it. Each new input multiplies the old residue by 2 and then adds either 0 or 1.



We now define the context-free grammar with the following production rules.

- $S \rightarrow 0Q_0$
- $S \rightarrow 1Q_11$
- $Q_0 \rightarrow 0Q_0$

- $Q_0 \rightarrow 1Q_11Q_0$
- $Q_0 \rightarrow \epsilon$
- $Q_1 \rightarrow 1Q_01Q_1$
- $Q_1 \rightarrow 0Q_20Q_1$
- $Q_1 \rightarrow \epsilon$
- $Q_2 \rightarrow 1Q_2$
- $Q_2 \rightarrow \epsilon$

Here S is the initial variable, Q_0 , Q_1 , and Q_2 are variables, and $0, 1$ are terminals. The production rules are designed to leave us in state q_0 , and the variables, roughly speaking, decay into cycles.

Problem 3. An *arithmetic expression* is an expression A of one of the following forms.

- a) $A = 0$
- b) $A = \omega$, where $\omega \in 1 \circ \mathbb{Z}_2^*$ (numbers are given base 2 and have no leading 0's)
- c) $A = -A_1$, where A_1 is an arithmetic expression
- d) $A = (A_1)$, where A_1 is an arithmetic expression
- e) $A = A_1 + A_2$, where A_1 and A_2 are arithmetic expressions
- f) $A = A_1 - A_2$, where A_1 and A_2 are arithmetic expressions
- g) $A = A_1 * A_2$, where A_1 and A_2 are arithmetic expressions
- h) $A = A_1/A_2$, where A_1 and A_2 are arithmetic expressions (note that while division by 0 is undefined, it is not invalid)

Let \mathcal{A} be the set of all arithmetic expressions. Give a context-free grammar whose language is \mathcal{A} .

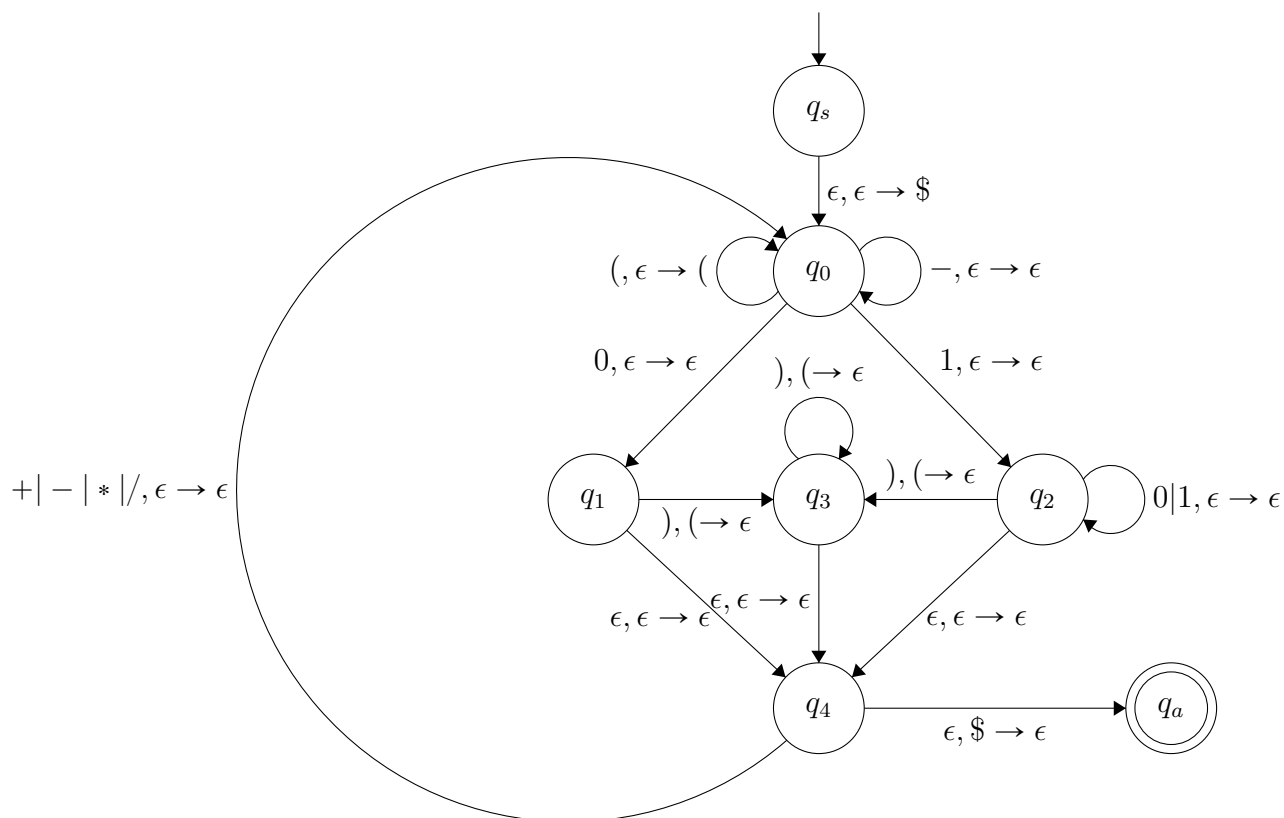
Solution 3. We define the context-free grammar for \mathcal{A} as follows.

- $A \rightarrow 0$
- $A \rightarrow 1\omega$ such that $\omega \in \{0, 1\}^*$
- $A \rightarrow -A$
- $A \rightarrow (A)$
- $A \rightarrow A + A$
- $A \rightarrow A - A$
- $A \rightarrow A * A$
- $A \rightarrow A/A$

Here A is the initial variable, and $+$, $-$, $*$, $/$, $($, $)$, 0 , and 1 are all terminals.

Problem 4. Draw a pushdown automaton which accepts the language \mathcal{A} .

Solution 4. First, note that we *always* alternate operand, operator, operand, and so on. The only exception to this pattern is the unary $-$ operator, which we may ignore since we don't need to actually evaluate the expression. As such, the only information we need to place onto our stack is how many parentheses we've encountered. We must match left parentheses with right parentheses. This gives us the following PDA.



Note that if we reach state q_a while we still have unprocessed input, the next input will cause the PDA to reject. The magic of nondeterminism will ensure we only transition to q_a when we actually wish to.

Problem 5. Give a context-free grammar G_n for each integer $n \geq 0$ with alphabet \mathbb{Z}_{n+1} (note that even when $m \in \mathbb{Z}_{n+1}$ is $m \geq 10$, m is a single symbol) such that the language of the grammar G_n is $0^*1^* \dots n^*$.

Solution 5. Let $n \geq 0$ be an integer. We define the context-free grammar G_n with the following production rules.

- $S \rightarrow A_1 \dots A_n$
- For each $i \in \mathbb{Z}_{n+1}$, $A_i \rightarrow iA_i$.
- For each $i \in \mathbb{Z}_{n+1}$, $A_i \rightarrow \epsilon$.

Here S is the initial variable, each A_i is a variable, and each $i \in \mathbb{N}_{n+1}$ is a terminal.